AWERProcedia
Information Technology
&
Computer Science

1 (2012) 782-787

2nd World Conference on Information Technology (WCIT-2011)

Software process improvement in small software development firms

Mohd Syazwan Abdullah*[a]*, Mejhem Yousef al-Tarawneh[b], Abdul Bashah Mat Ali[a]

[a]School of Computing, College of Arts and Sciences,Universiti Utara Malaysia, 06010 UUM Sintok, Kedah, MA LAYSIA

Abstract

Most of small software development firms use ad-hoc manner in developing their software products where it is difficult for them to achieve the requirements of the traditional software process improvement (SPI) models and standards, as these traditional models and standards were developed for large firms. As such, these firms are suffering from the lack of suitable software development methods that will enable them to follow the SPI models. This can be addressed by having a suitable software development process improvement framework to enable them in delivering high quality software products that fulfil the customer's expectation faster and cheaper. This paper presents the software development process improvement framework which was developed based on the Capability Maturity Model Integration Version 1.2 for Development (CMMI-DEV1.2) as a software improvement model and eXtreme Programming (XP) as a software development method. It discusses the development processes of this framework by highlighting the steps and stages involved.

Keywords: Software Development Process Improvement Framework; XP method, CMMI-DEV1.2; Small Software Development Firms.

Selection and/or peer review under responsibility of Prof. Dr. Hafize Keser.

1. Introduction

These days, the software industry forms the most rapidly growing economic sectors, where the small software development firms play a fundamental role in most countries economies [1]. Nevertheless, these firms need to manage and improve their software development activities to satisfy the customer's needed within the time constraints at lower cost, while maintaining high quality. SPI models can be used as a competitive advancement strategy for both small and large firms [2]. However, small software development firms do not have suitable development and management processes to follow the current SPI models. Most of the small software firms use

* ADDRESS FOR CORRESPONDENCE: Mohd Syazwan Abdullah, School of Computing, College of Arts and Sciences,Universiti Utara Malaysia, 06010 UUM Sintok, Kedah, MALAYSIA

informal or ad hoc manner to develop their software products. Furthermore, all of SPI models were developed for large firms and they are too complicated and expensive to be implemented by small firms [2].

Despite the fact that SPI traditional models are unsuitable for small software firms; however adapting SPI in these firms is still possible by simplifying the way of using these improvement models [3]. From this point, small software firms need a lightweight software development method to develop their software products in a systematic way, and appropriate SPI model to be followed in order to mange and improve their software development processes [3][4]. As such, the XP method and CMMI-DEV1.2 model have been chosen as a software development method and SPI model in developing a suitable software development process improvement framework for these firms. This framework aims to help small software firms in managing and improving their software development activities to be able to deal with the rapid technology advances, maintaining their products, satisfying the customer's needs and sustaining the firm. This paper discusses about SPI in small software development firms; CMMI-DEV1.2; and XP method. Furthermore, this paper also presents the development processes of the software development process improvement framework for small software development.

2. SPI in Small Software Development Firms

Small software development firms represent the fundamental growth of many national economies and it is important to notice that the contribution from these firms should be seen as important and significant as compared to the large ones [1], as they develop a large portion of the needed software applications, offer many job opportunities, and exploit new technologies. However, small software firms have different characteristics compared to other sizes of software firms such as number of employees, capital, recourses, tools used, methods adopted, and other features [5]. Cater-Steel [6] summarized the characteristics of small software firms compared with large software firms as following:

- These firms have a flat structure, less formalized decision-making structures and procedures, and provide more freedom for employees to depart from the rules.
- In these firms; just one or two person has\have the critical management decisions in finance, accounting, personnel, purchasing, processing or servicing, marketing and selling.
- These firms have the features that enable them to be responsive and flexible.
- These firms neglect the training compared with larger software firms.

Given the importance of small software development firms all over the world; there is a need to help them for managing and improving their software development activities to keep them alive. Unfortunately, most small software firms could not afford to follow the SPI traditional models and standards [2]. In addition, these firms suffer from the lack of understanding in the success factors of SPI and do not have enough people to perform all the SPI activities [7]. Thus, they find themselves to be very far from implementing formal SPI traditional models and standards, and also perceive these models as expensive and time consuming [6]. The problems faced by the small software firms in implementing the SPI traditional models which are: financial problems; undefined organizational structure and responsibilities; organizational success is based on individual skills; and long term return on investment [4]. Furthermore, small software development firms suffer from the lack of research studies to solve their problem of improving their software processes by the SPI traditional models, as most these studies were focused on large and very large firms [6] [3]. This warrants the need to conduct research on these firms for implementing the suitable SPI model capable of managing and improving their software development activities. For this reason, the present study aims to help these firms by developing a suitable software development process improvement framework to enable them in managing and improving their current software development activities.

3. CMMI-DEV1.2

CMMI-DEV 1.2 [8] is a process improvement maturity model which was developed especially for software development industry to provide a comprehensive integrated solution for development and maintenance activities related to products and services. This model consists of best practices that address development and maintenance activities that cover the product lifecycle from conception through delivery and maintenance. In addition, CMMI-DEV1.2 supports two improvement paths which are: (1) continuous representation: applied to an organizations process improvement achievement in individual process areas; and (2) staged representation: applied to an organization's process improvement achievement across multiple process areas. Furthermore, both representations have the same twenty two key process areas, which provide ways to implement process improvement to achieve business objectives and provide the same essential content as well as have the same key process areas; however each one has special strategy in improving the software process. Furthermore, CMMI-DEV 1.2 is considered as one of the best known model which focuses on SPI for achieving quality software in small software firms [2]. In addition, this model can aid small software firms to achieve their quality goals when used as guidelines for software process improvement [9]. Thus, CMMI-DEV1.2 is used in this study as a comprehensive improvement model in developing the software development process improvement framework for small software development firms.

4. Extreme Programming(XP)

XP is the most popular and effective agile method [10]. It is called extreme because it takes the things that are good in software development and applies them in an extreme manner. This method has been introduced as a result of the long development cycles of tradition and conventional development models. XP proposes twelve software development practices to increase productivity, while maintaining quality which are: planning game; metaphor; simple design; test-driven development; design improvement (refactoring); pair programming; collective ownership; continuous integration; sustainable pace (formerly 40-hour weeks); on-site customer; and coding standards. In addition, there are several different roles with different tasks that are used during the software process of XP which are: programmers; customer; tester; tracker; coach; consultant; and big boss. Given the suitability and effectiveness of XP method for small software development firms and the high coverage of XP practices to the specific goals of CMMI-DEV1.2 key process areas [11], the XP method has been used as a software development method in developing the new framework.

5. Developing the Software Development Process Improvement Framework for Small Software Development Firms

There are four steps used to develop the software development process improvement framework for small software development firms which are [11]:

- Comparing XP method to CMMI-DEV1.2.
- Extending XP method to fulfil the missing CMMI-DEV1.2 key process areas.
- Developing the proposed software development process improvement framework for small software development firms based on the proposed Extended-XP method and CMMI-DEV1.2.
- Verifying and refining the proposed framework by using focus group method.

In the first step of the development strategy, XP practices [10] and the specific goals of each CMMI-DEV1.2 key process area [8] have been used as the main items in extracting the adjustments and conflicts between XP and CMMI-DEV1.2. As a result of this comparison [11], the missing key process areas were divided into three groups which are: largely supported; partially supported; and not-addressed key process areas. In the second step, the required software development, management, and improvement additions which are needed to fulfil the partially and not-addressed key process areas were extracted from the related literatures to extend the XP method. The Extended-XP method is discussed in section 5.2. As a result, the Extended-XP method had been

used as a software development method in the proposed framework. The foundation of the proposed framework is in [11].

In general the SPI framework [12] consists of four generic elements which are: software process; assessment; capability determination; and improvement strategy. These elements must be used as a baseline in developing the proposed framework. Thus, there is the need to rearrange these elements to be suitable for software development and improvement issues by integrating the CMMI-DEV1.2 model and the proposed Extended-XP method as a generic elements in the proposed framework. In this aspect, several changes have been done to the contents of the generic elements of SPI framework because the proposed software development process improvement framework focuses on development and improvement issues. In this foundation, the Extended-XP has been used as a software development method instead of the improvement strategy, while the CMMI-DEV1.2 model has been used as an assessment model in the proposed framework. Based on this, the proposed framework has been developed and was then presented to focus group members to verify the compatibility of this framework to the key process areas of CMMI-DEV1.2 taking into account the characteristics of small software firms. As a result of the focus group method, several modifications had been made to the proposed framework and the proposed Extended-XP method. The major modification done was in removing the related activities of the organizational and innovation deployment process area from the proposed framework, because focus group members argued that this area is not suitable for small software firms.

As for the roles of the software development process improvement framework; this framework consists of the same roles of XP method with several additional practices to programmers, coach, and tracker. Furthermore, there are two new roles that have been added to this framework compared to XP roles which are framework-SEPG members and Extended-XP-SEPG members [11]. Thus, the framework roles consist of the following: programmers, customer, tester, coach, tracker, consultant, big boss; Extended-XP-SEPG members; Framework-SEPG members. Sections 5.1 and 5.2 discuss the stages of the verified framework and the new phases of the verified Extended-XP method.

5.1. Framework's stages

As a result of the focus group verification, the software development process improvement framework consists of three stages which are [11]:

- Stage one: assessing the current software development processes
  Before implementing the framework, it is important to determine the suitable simple project repository by framework-SEPG members to keep the important data during the implementing of this framework. Here, Microsoft Office is suggested as a free tool for data storing issues. Then, the framework-SEPG members are responsible for assessing the current capability levels of these processes by using CMMI-DEV1.2 key process areas. In this aspect, three scales can be used to identify the capability levels of the current development processes which are: largely supported; partially supported; and not supported. As results of the current levels, framework-SEPG members are responsible for modifying and rearranging the current software development activities to be suitable with the required roles of the verified framework. This can be done by distributing the new roles of the verified framework to the project team members that are related with their experiences. At the end of this stage, the new roles are known for each employee in the firm.
- Stage two: adopting Extended-XP method
  In order to implement the Extended-XP method in the right way; all of the involved team in the software development processes must have a good knowledge of their roles and they must be trained. The best way to learn the Extended-XP method is through training courses. Furthermore, there is a need to support the team with the required XP books and with the documentation of the Extended-XP method during the training and the development lifecycle. In this case, Extended-XP-SEPG members are responsible for carrying out the training process before implementing the Extended-XP method by establishing plans for training the developers; estimating the time required of training; training the developers on how they can implement

the activities of the Extended-XP; and assessing the project team's efficiency. As results of the training process, the project team will be ready and can start adopting the Extended-XP method in the right way.

- Stage three: identifying the best practices of the current project
  Based on the results of the second stage, framework-SEPG members are responsible for meeting the project team to discuss the best practices of implementing the verified framework by using the specific practices of CMMI-DEV1.2 key process areas as main items in this discussion. In this questionnaire; four choices have been used to answers CMMI-DEV1.2 questions which are: "Yes", "Don't know", "Does Not Apply", and "No". However, as results of this meeting, the questions can extract the best practices of implementing the verified framework for the current project. Finally, the framework-SEPG members are responsible for keeping these best practices in the project repository and will be taken into account for later projects.

5.2. Extended-XP method

The typical XP method [10] consists of six phase which are: exploration, planning, iteration to release, productionizing, maintenance, and death. However, the Extended-XP method [11] only consists of four phases which are:

- Requirement management phase: this phase has all the activities of the exploration and planning phases of XP method [10]. In addition, there are new process that was added between the exploration and planning processes which are responsible for supporting the current project with the unavailable required development tools or services [10].
- Development phase: This phase has the same activities of iteration to release phase in the XP method [10].
- Product delivery and product & process efficiency phase: this phase consists of the same activities of productionizing phase in XP method [10] and other additions that are needed to support the quality assurance of process and product to achieve the organizational process performance. In addition, there are several metrics that could be appropriate for objectively verifying the products and the process such as: (1) release plan adherence, (2) percentage of test cases that are running successfully, (3) percentage of acceptance tests that are running successfully, (4) length of pair programming sessions, and (5) project velocity. Furthermore, it's important to convey the metrics through defined channels to the affected parties and senior management.
- Maintenance & death phase: This phase consists of the same activities of maintenance and death phase of XP method [10].

6. Conclusion

Small software development firms are larger in number compared to larger software development firms all over the world and play a fundamental role in most countries economies. However, these firms need to manage and improve their software development activities to be able to deal with the rapid technology advances, maintaining their products, and satisfying the customer's needs. However, all SPI traditional models were developed for large firms, therefore small software firms find themselves to be very far from implementing these models. Furthermore, there are four general problems faced by small software firms in SPI which are financial problems, undefined organizational structure and responsibilities, organizational success is based on individual skills, and long term return on investment. In addition, there is a lack of research studies to solve the problem of improving their software process by the SPI traditional models. This study aimed to help small software development firms in developing a suitable software development process improvement framework for these firms to enable them for managing and improving the software development activities.

References

[1]. S. B. Basri, and R. V. O'Connor, "Knowledge Management in Software Process Improvement: A Case Study of Very Small Entities," *Knowledge Engineering for Software Development Life* Cycles: Support Technologies and Applications, p. 273, 2011.

[2]. I. Garcia, C. Pacheco, and J. Calvo-Manzano, "Using a web-based tool to define and implement software process improvement initiatives

in a small industrial setting," Software, IET, vol. 4, pp. 237-251, 2010.

[3]. H. Altarawneh, and S. Amro, *Software Process Improvement In Small Jordanian Software Development Firms*. Paper presented at the 7th International Conference on Perspectives in Business Informatics Research (BIR'2008), Gdansk, Poland. PP 175-189. 2008.

[4]. M. Zarour, Methods *to evaluate lightweight software process assessment methods based on evaluation theory and engineering design principles*. Doctoral's Thesis. Computer Science Departement, Du Quebec Uni, Canada, (2009).

[5]. C. Hofer, Software *development in Austria: results of an empirical study among small and very small enterprises*.Paper presented at the 28th Euromicro Conference (EUROMICRO'02). Dortmund, Germany, 2002.

[6]. A. P. Cater-Steel, *"Low-rigour, rapid software process assessments for small software development firms,"* Paper presented at the Australian Software Engineering Conference (ASWEC'04), Australia.2004, pp. 368-377.

[7]. F. Guerrero & Y. Eterovic, Adopting the SW-CMM in a Small IT Organization. *IEEE software. 21*(4), pp. 29-35, 2004.

[8]. CMMI Product Team, CMMI for Development, version 1.2. Preface, Software Engineering Institute, Carnegie Mellon University, USA, 2006.

[9]. G. V. Boas, A. R. C. da Rocha, and M. Pecegueiro do Amaral, *An Approach to Implement Software Process Improvement in Small and Mid Sized Organizations*. Paper presented at the Seventh International Conference on the Quality of Information and Communications Technology. Porto, Portugal.

[10]. K. Beck, *Extreme programming explained: embrace change*: 3th End.Reading, Mass, addition-Wesley. Boston, 2000.

[11]. M, Al-tarawneh, S*oftware development process improvement framework for small software development firms*, Doctoral's Thesis. School of Computing, College of Arts and Science, Universiti Utara Malaysia, Malaysia, Unpublished.2012

[12]. R. Pressman, *Software Engineering: A Practitioner's Approach.* 7th international edn, McGraw-Hill Education, Newyork, USA, 2009.